Vol. 13 No. 2
Apr., 2016

天 文 研 究 与 技 术
ASTRONOMICAL RESEARCH AND TECHNOLOGY

第 13 卷 第 2 期
2016 年 4 月

CN 53-1189/P ISSN 1672-7673

基于 ROACH2-GPU 的集群相关器研究*
——X-engine 模块的设计与实现

汪群雄[1,2]，牛晨辉[2,3]，田海俊[1,2]，吴锋泉[2]，李吉夏[2]，陈学雷[2]，蒿 杰[4]

(1. 三峡大学，湖北 宜昌 443002；2. 中国科学院国家天文台，北京 100012；3. 华中师范大学，湖北 武汉 430079；4. 中国科学院自动化研究所，北京 100190)

摘要：随着射电干涉技术的不断提升，干涉阵列规模越来越大，观测能力逐渐增强，但随之而来的是超大数据的实时处理问题。针对该问题，结合射电干涉仪相关器在数据运算和传输等方面的需求以及射电干涉阵列信号的特征，研制了一套基于图形处理器集群的通用相关器并用于“天籁计划”的数据处理：首先根据射电信号的关联计算特性，按频段将计算任务分配到不同图形处理器节点，并合理均衡各节点网络负载；然后由不同图形处理器节点独立完成各自的计算任务并将计算结果实时送往存储节点；最后按图形处理器集群通用相关器的设计方案成功安装部署系统并根据“天籁计划”一期的需求进行了性能测试。该图形处理器集群相关器计算性能约为理论峰值性能的 46%；相对于传统方案的相关器，基于图形处理器集群的相关器具有开发周期短、可扩展性强、部署简单等优势。

关键词：射电干涉仪；图形处理器相关器；图形处理器集群；数据实时处理；分频式计算

中图分类号：P111.47 **文献标识码**：A **文章编号**：1672-7673(2016)02-0219-09

时间序列信号模数转化、干涉显示度的计算和校准以及噪音的消除和天图的傅里叶重构是射电干涉阵信号处理的主要过程，其中干涉显示度的计算是最关键也是计算量最大的部分，该部分由相关器完成，它主要包括各路信号的傅里叶变换(F-engine)和交叉互关联(X-engine)。干涉显示度的计算量随射电干涉阵的阵元数目平方的增长而增长[1]。目前国际上射电干涉阵列的阵元个数已达数百乃至数千，其信号的实时处理需求已趋于万亿次每秒甚至亿亿次每秒。例如 2013 年投入运行的“阿塔卡玛毫米/亚毫米波阵列望远镜”(the Atacama Large Millimeter Array, ALMA)，拥有 66 面天线；我国的“天籁计划”项目[2]，一期规模达 96 个双极化阵元，现已基本组建完成，二期计划建近千个阵元；国际上多国合作正在筹建的平方千米阵列望远镜(Square Kilometer Array, SKA)第一期将包括由约 200 个碟形天线组成的中频阵以及由约 50 个基站、13 万个阵子天线组成。如此大规模的天线阵列，对数据采集、传输以及实时处理都将带来巨大的挑战，如何应对这些挑战是国际上目前关注的一个难题[3]。

为了解决干涉阵数据实时处理问题，传统方案采用硬件专用集成电路(Application Specific Integrated Circuit, ASIC)或现场可编程门阵列(Field-Programmable Gate Array, FPGA)设备进行射电信号的交叉关联，通常这种方式开发周期长、可扩展性差、部署困难且费用高。针对天籁实验干涉阵研发了一套具备开发周期短、可扩展性强、部署简单和费用低廉等优势的通用相关器，基于图形处理器集群设计了一套解决方案。图形处理器不仅具备高性能和高质量的图形处理能力，同时更具有杰出的浮点计算能力以及极高的存储器带宽，这些特性使得图形处理器在射电干涉仪相关器的研发上具有极大的潜力。

本文介绍了采用图形处理器统一设备计算架构(Compute Unified Device Architecture, CUDA)设计软件模型并结合硬件完成的相关器的开发。软件设计相对硬件设计而言有开发周期短、可扩展性强、部署简单和费用低廉等优势，因此使用软件实现射电信号的交叉关联替代硬件实现是一个很有价值的

* 基金项目：国家自然科学基金 (U1231123, 11503012, U1331202, U1431108)；863 科技攻关计划 (2012AA121701) 资助.
收稿日期：2015-12-28；修订日期：2016-01-22

作者简介：汪群雄，男，硕士研究生. 研究方向：高性能计算. Email：1276303919@qq.com

解决方案。在面对未来更大规模射电望远镜的实时信号处理时，该相关器只需修改相应参数便可实现灵活扩展，通用性强。在之前的实验中[4]，对单图形处理器的计算性能进行了测试，证实了图形处理器的强大潜能，但随着阵元数目的增加，单图形处理器很难满足干涉阵数据计算的实际需求，因此需设计图形处理器集群下的交叉关联算法，通过图形处理器集群强大的数据处理能力来弥补单图形处理器的不足。

1 基于图形处理器集群的相关器设计

针对射电干涉阵采集的密集型观测数据，图形处理器集群相关器在数据实时处理过程中存在两个关键性问题：(1)数据的合理分发与调度；(2)集群节点运算性能的优化。当数据分发调度出现问题时，势必会导致各节点的负载不均衡，影响各节点的运算性能；反之，节点运算性能不佳，也会影响数据的整体调度。二者相互依赖，相互制约。图形处理器集群相关器的架构设计需要同时兼顾这两个核心问题，才能使集群相关器整体性能达到最佳。为解决第一个问题，采用了分频分布式计算的信号处理模式并借鉴 CASPER(美国 Berkeley 一家天文信号处理与电子研究合作组织)的相关器技术①，实现数据和任务的分发与调度；针对第二个问题，在之前的实验中已经尝试了多种图形处理器优化方法。

1.1 图形处理器相关器的整体架构设计

由于先做傅里叶变换会将时域信号变成频域信号，再做交叉关联时，不同频率之间的相关为 0，所以只需要做同频率关联②，故而 FX 方式较 XF 方式计算复杂度小，所以设计 FX 相关器做数据的实时处理。图 1 是图形处理器集群相关器的整体设计框架。

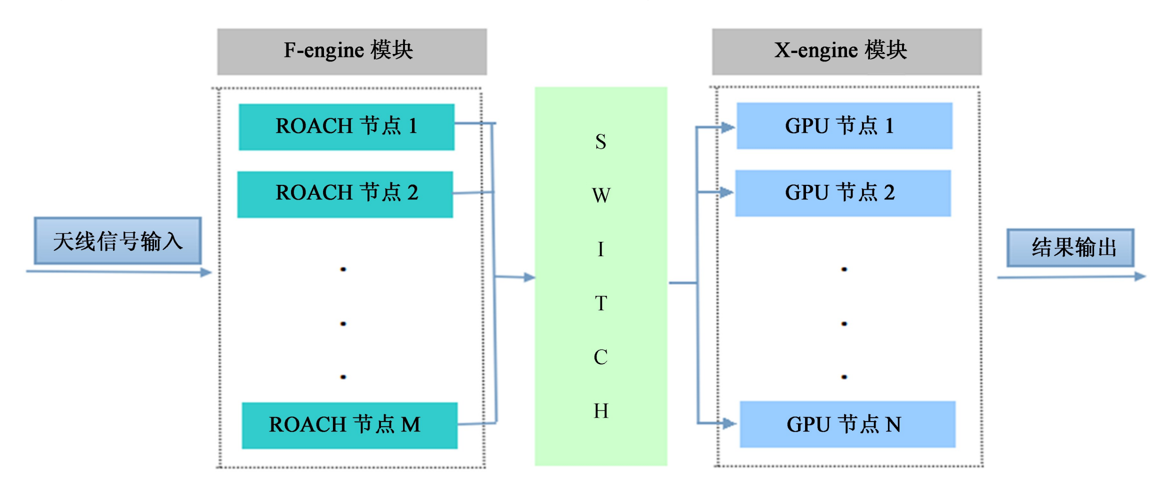


图 1 图形处理器集群相关器框架图

Fig. 1 The architecture of the GPU-cluster-based correlator

相关器硬件平台采用由美国 CASPER 研究组开发的具有开放式可重构架构的 ROACH 服务器，如图 1。F-engine 模块运行在 ROACH 服务器节点上，主要通过 ROACH 板(由 CASPER 研制的一种现场可编程门阵列处理板)做数据采样及快速傅里叶变换等相关操作，本文着重讲述 X-engine 模块的设计实现；X-engine 模块在图形处理器服务器上运行，该部分主要负责数据的交叉关联并输出结果。连接 F-engine 模块与 X-engine 模块的是网络交换机。通过交换机实现数据的合理分发与调度。

假设图形处理器集群相关器 F-engine 模块含有 M 个 ROACH 服务器节点，X-engine 模块有 N 个图形处理器服务器节点；并且每个 ROACH 节点对应 m 个天线的数据采样，经过快速傅里叶变换后，则对应 m 路频域信号；每路信号有 F 个频点，则每个图形处理器节点对应于 F/N 个频点，每个频点所

① Collaboration for astronomy signal processing and electronics research，https://casper.berkeley.edu/wiki/PAPER_Correlator_Manifest

② Radio astronomy tutorial. Internet，http://www.haystack.mit.edu/edu/undergrad/materials/ RA_tutorial.html

有 $M*m$ 路信号的交叉关联计算。即将每路信号作快速傅里叶变换处理后的 F 个频点按前后顺序分为 N 段，每段有 F/N 个连续频点，其中，每个图形处理器节点负责某个固定频段的数据关联计算，所以每个 ROACH 节点需要将某特定频段通过交换机送往对应的图形处理器节点；而在图形处理器节点内部，根据其拥有的图形处理器计算核心个数 n，再一次将频点均匀地分为 n 段交由 n 个图形处理器核心做关联计算。X-engine 模块的详细结构见 1.3 节。

1.2　分频分布式计算模式

根据干涉阵数据交叉关联过程按同频相关的特性(不同频率的相关性为 0)，采用分频分布式处理方案。该方案具体操作：不同频段的相关计算被分配到不同的节点上。假设共有 N 个频率，分别被分配在 N 个单元上计算。其中，1 单元负责计算所有天线馈源对之间频率 1 的干涉显示度，因此余下 N-1 个单元都将频率 1 的数据发送到 1 单元；2 单元负责计算所有馈源对之间频率 2 的干涉显示度，则其余 N-1 单元都将频率 2 的数据发送到 2 单元；依此类推。这样，每个单元得到且仅得到它负责计算的那个频率的所有数据。假定每个单元的数据采集速度为 B，由于共有 N 个频率，因此从其它每个单元得到数据的速率为 B/N(这里略去了数据打包时头文件的数据量)，共有 N 个节点，因此从其它节点得到的总数据流量为 $B(N-1)/N$。同样，从该单元传给其它单元的数据流也是这么多。如图 2。

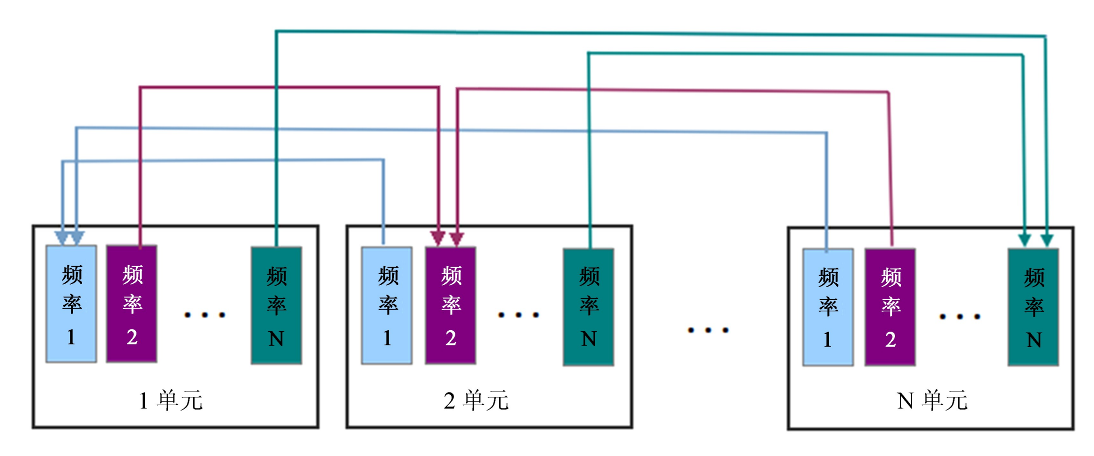


图 2　分频分布式计算架构

Fig. 2　The architecture of distributed computing

图 2 中，频率个数与计算单元数相等，这只是为了方便画图说明；在实际情况中，二者通常并不会相等，即存在一个单元负责多个频点数据相关计算的情况，这样才能充分体现分频分布式计算模式的优点，大大降低数据的传输压力和数据交换的复杂度[5]。

1.3　集群环境下 X-engine 模块设计

由于在相关器数据处理时，大部分的计算量集中在 X-engine 模块，所以下面重点介绍该图形处理器集群相关器的 X-engine 模块的详细设计结构(图 3)。

在图 3 中，图形处理器节点之间构成分布式，即将实际的频点计算任务按频段均衡地分发到所有节点上进行计算；各计算节点采用中央处理器+图形处理器主从结构，即是节点内异构式。中央处理器负责与当前节点进行交互，收发命令和数据，并控制图形处理器进行计算；中央处理器单元和图形处理器单元内部共享内存空间和显存空间，采用内存统一寻址[6]。

X-engine 模块采用 CUDA C 语言实现，主体程序分为串行部分和并行部分，并行部分又分为集群节点间并行和节点内线程间的并行。串行部分由中央处理器执行；到并行部分，将任务初步分配到各图形处理器节点，在节点内再对任务进一步划分，并将划分好的任务送到协处理器图形处理器上进行计算；在图形处理器完成并行计算任务后，将结果由图形处理器拷贝到内存，并由中央处理器输出到指定的后端存储设备；至此，一个积分时间内的所有数据传输与计算任务完成。

1.4　相关器数据分发与调度

CASPER 在相关器实现过程中，并未将 F-engine 和 X-engine 两部分集中在一个模块，而是将其分

开实现并采用网络交换机将二者连接，主体架构类似文中图 1；这样，根据各部分的实际计算能力以及传输速率便可以合理地进行数据的分发与调度。相关器设计也采用了这一方案，从而实现了 F-engine 和 X-engine 模块的分离，可以根据实际的数据计算量灵活地增加或减少任意模块中计算节点的个数，有效增强了相关器的可扩展性，从而具备较强的通用性。

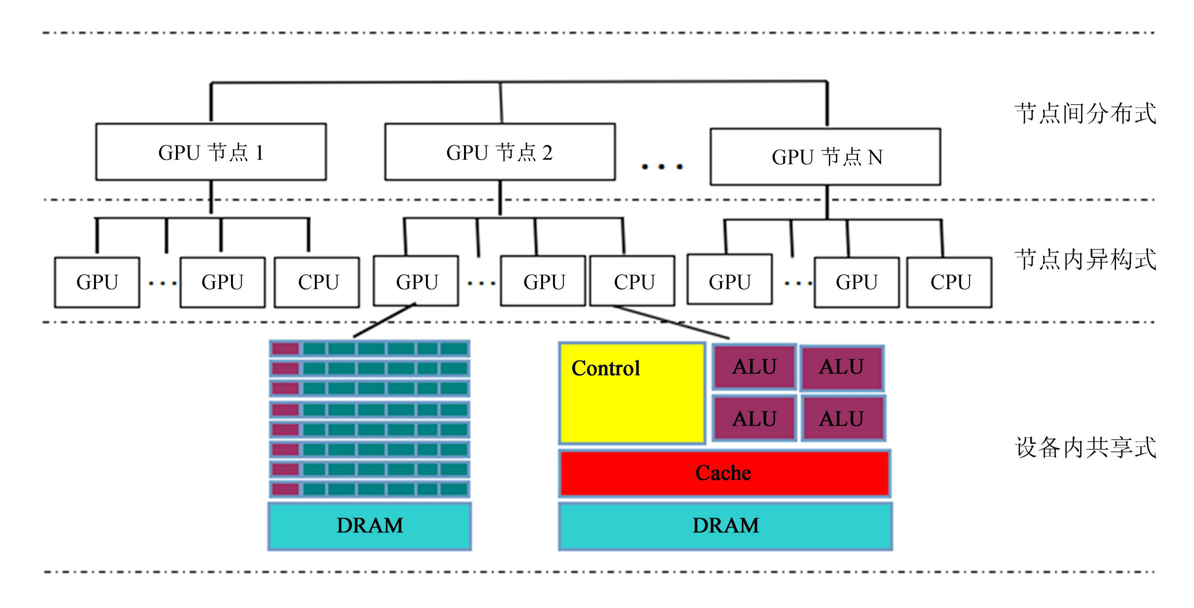


图 3 X-engine 模块结构

Fig. 3 The structure of X-engine module

2 图形处理器集群相关器的硬件实现

设计了一套基于图形处理器集群的相关器，并根据“天籁计划”实际的计算任务予以硬件实现。

2.1 节点设计

根据相关器的构架，实现该相关器的关键在于确定 F-engine 模块和 X-engine 模块中节点的个数。根据计算性能的需求，可以确定图形处理器节点的数量。首先，实际上每个图形处理器节点具有 4 个图形处理器计算核心(视节点功耗、主板插槽、空间等情况而定)；经过反复测试，每个图形处理器计算核心的实际计算性能约为 1.2TFLOPS(峰值性能)，所以每个图形处理器节点最佳计算性能是 4.8TFLOPS，“天籁计划”一期规模总计算量为 35.6TFLOPS[4]，则 X-engine 模块至少需要 8 个图形处理器节点才能完成计算任务。对于 F-engine 模块，由于 F-engine 阶段的数据计算量较小，故只需要满足数据传输要求即可。

在数据传输方面，采用万兆网卡，为了确保在程序运行过程中不至于因网络传输满负荷而导致数据丢失，测试时的平均网络传输速率维持在 0.8 GB/s 左右；程序在 F-engine 传输数据之前需对原始信号进行截位操作，将 10 位的原始数据截取 4 位有效数据；则由总数据输入速率可知，要完整地接收所有数据，至少需要 24 块万兆网卡。实际上，每个 ROACH 服务器和图形处理器服务器上均插有 4 块万兆网卡；即按照数据传输要求，F-engine 模块和 X-engine 模块至少分别需要 6 个 ROACH 节点和 6 个图形处理器节点。

综合数据传输以及计算性能需求考虑，相关器 F-engine 模块的节点数量应为 6，X-engine 模块的节点数量应为 8。

2.2 相关器各模块的实现

对于 F-engine 模块的实现，在此只作简要说明。相关器 F-engine 模块节点个数为 6，即 6 台 ROACH 服务器，每台服务器通过 ROACH 板采集数据，经过截位、快速傅里叶变换等一系列处理后，由 4 个万兆网口经交换机送往 X-engine 节点。

X-engine 模块需要 8 个图形处理器节点，即对应 8 台图形处理器服务器，每台服务器配置 4 个图形处理器计算核心(2 块 GTX690，每块 2 个计算核心)、4 个万兆网口以及至少 12 个中央处理器核心；每个图形处理器核心处理固定的 32 个频点数据，4 个图形处理器核心分别独立并行执行。除基本硬件配置外，重点是数据处理的软件实现。图 4 是图形处理器节点内部程序实现的架构，反映了每个图形处理器节点内部程序实现的线程、缓冲区以及数据流向(箭头所示)之间的关系。

从图 4 可以了解到 X-engine 模块中各图形处理器节点数据处理的软件实现框架结构，即该数据处理程序由 4 个主线程组成。其中，网络线程负责接收来自 ROACH 服务器的数据并将其按频点先后、天线顺序等特定要求重新整合，该过程数据存放在图形处理器缓冲区内。图形处理器缓冲区有 4 个缓冲数据块，当某个数据块满足交叉关联的要求，则将该数据块拷贝到图形处理器显存然后清空该数据块所占内存空间；图形处理器线程负责数据的交叉关联计算，并将积分结果拷贝到中央处理器缓冲区；中央处理器线程则负责对中央处理器缓冲区内的数据做结构调整，并将最后结果存放在硬盘缓冲区内；硬盘线程专门负责将最后的计算结果送到后端的存储设备。在整个过程中，程序开辟了一个状态变量缓冲区，具体作用是实时获取各线程以及缓冲区的状态信息，通过读取并显示该缓冲区的变量便可实现程序运行状态的实时监控。

借鉴 CASPER 的相关器设计，X-engine 模块在数据处理时，对输入的数据包有特定的数据结构要求，这便于后续的数据整合处理。图 5 是 X-engine 模块图形处理器节点内网络线程要求输入的数据包格式。

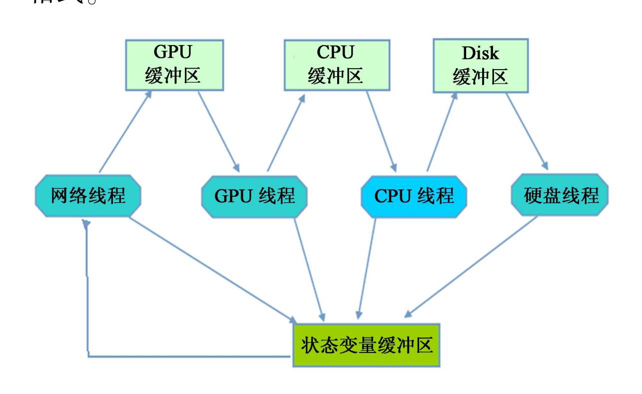


图 4　图形处理器节点内部程序架构

Fig. 4　The program architecture within a GPU node

Head			MCNT						FID	XID
Payload	t0	ch0	inA	inB	inC	inD	inE	inF	inG	inH
		ch1	inA	inB	inC	inD	inE	inF	inG	inH
		…	.							
		…	.							
	t1	ch0	inA	inB	inC	inD	inE	inF	inG	inH
		ch1	inA	inB	inC	inD	inE	inF	inG	inH
		…	.							
		…	.							
	.		.							
	.		.							
	.		.							

图 5　网络数据包格式

Fig. 5　Network packet format

由图 5 可知，网络线程接收的数据包头部含有 3 个参数 MCNT、FID 和 XID，分别表示当前包序列号、当前包来源于 F-engine 模块的哪个节点以及当前包被送往 X-engine 模块的哪个节点。Payload 部分是具体的数据信息，t0、t1 等表示采样次数；ch0、ch1 等表示信号路数。每次采样中，每路信号采集 8 条频谱；每个数据包大小(包含的采样次数和信号路数)可根据实际传输情况进行适当的调整。文中实现的图形处理器集群相关器要求数据包大小(不含包头部大小)为 8 192 字节。

3　图形处理器集群相关器的性能测试

3.1　相关器误差分析及正确性验证

该图形处理器相关器的误差来源主要有两部分：(1)F-engine 模块中的信号采样、截位等过程；(2)X-engine 模块中的交叉关联过程。对于(1)过程的误差另文说明，现主要分析(2)过程的误差。X-engine 模块中的交叉关联在图形处理器中完成，其实质是浮点数的乘累加过程。由于图形处理器的单精度浮点数计算性能远高于其双精度浮点的计算性能，为充分利用这一优势，程序要求输入的数据采用 32 bit 的单精度浮点数类型，该类型在计算机中的二进制存储分为 3 部分：符号位(1 bit)、指数位(8 bit)以及尾数位(23 bit)；所以，对于单个结果的最佳精度为 1×2^{-23}，在 1×10^{-7}到 1×10^{-6}之间；而

在累加过程，浮点数加法运算需要进行对阶和右规范化操作，该操作会进行舍入处理而造成误差，误差随累加过程不断积累，为尽量消除误差，程序采用分组相加方法[7]，最后经过整体测试，图形处理器程序在单精度浮点数乘累加过程中的计算结果与中央处理器采用双精度的计算结果的最大误差约为 1×10^{-12} 量级。

此外，为了检测该相关器的最终计算结果正确与否，对相关器输入模拟白噪声信号，通过加入两路延时检测相干相位随频率的变化，并与理论值进行比较。其中，相位随频率变化的理论值 K 为

$$K = 2\pi\Delta T \ , \tag{1}$$

式中，ΔT 表示时间延迟。在检测试验中，加入的时延为 $\Delta T = 2.5\times10^{-8}$ s，所以相位随频率变化的理论值 $K\approx1.571\times10^{-7}$；然而，根据实际观测数据拟合得到的相位随频率变化的大小 $\bar{K}\approx1.561\times10^{-7}$。所以该图形处理器集群相关器计算结果的相位随频率变化率与理论值的误差 ΔE 为

$$\Delta E = \frac{|K - \bar{K}|}{K} \ , \tag{2}$$

由(2)式可以得出，图形处理器集群相关器相位随频率变化率误差 $\Delta E\approx0.006365$。

最后，将干涉阵天线接收的信号分别送往一套基于现场可编程门阵列和数字信号处理(Digital Signal Processing，DSP)的相关器(该相关器由中科院自动化所研制)和上述图形处理器集群相关器进行计算，对二者的计算结果进行比较。如图 6，上、中、下 3 幅图分别是实验过程中两套不同相关器对同一段信号的计算结果的相位图以及二者的相位差。

将图 6 的上面两幅图作对比不难发现，这两套相关器对相同信号计算结果的相位图几乎完全一致；而在图 6 最后一幅相位差图中，可以进一步证实这一点，二者的相位差基本为 0，图中有些非零部分主要由于噪声所致。

3.2 图形处理器相关器计算性能及传输性能测试

此前实验中，针对 GTX460 和 GTX480 测试了不同天线情况下数据传输的速率以及计算性能。下面针对该图形处理器相关器的某一图形处理器节点，测试 GTX690 的数据传输性能和计算性能，分别如图 7(a)、(b)。

从图 7 的计算性能曲线图可以看出，GTX690 的内核性能在天线个数为 96 时达到最高，约为 1200GFLOPS，该性能约占理论峰值性能的 46%；而图形处理器节点的整体计算性能随天线个数递增。从数据传输速率曲线图中可以得出，对于图形处理器节点内的传输速率来说，其主要瓶颈在设备与主机之间的数据传输，即是 PCI-E 的传输速率限制，这与前期实验中的结论一致；然而，设备与主机间的传输限制在图形处理器集群中并非唯一的问题，因为从网络传输来看，每个图形处理器节点对应 4 个万兆网卡，实际的网络传输速率峰值约在 4 GB/s，而在相关器具体实现中，由于计算需求，图形处理器节点对接收的数据需先做移位操作然后才拷贝到图形处理器显存进行计算，移位操作将原来 4 位数据左移 4 位变为 8 位，相当于网络传输速率峰值变为 8 GB/s，而这个网络峰值传输速率与图 7 中测试的主机设备间的最大数据传输率相当，即对于该图形处理器相关器而言，数据传输受限于网络以及图形处理器节点内设备与主机之间的传输(PCI-E 传输速率)。

3.3 图形处理器相关器其它性能

在相关器的诸多性能中，可拓展性尤为重要。基于传统的相关器，即单纯采用硬件 ASIC 或 FPGA 设备来进行射电信号的交叉关联运算的相关器，由于受限于硬件，其计算性能、功耗等基本固定，当计算量规模发生变化，其可拓展性极差；而基于图形处理器集群的相关器，通过软件编程实现相关器性能与硬件的分离，不再完全依赖硬件。图形处理器相关器可以根据具体的计算任务做相应调整。例如，“天籁计划”阵列规模从一期的 96 面天线扩建到约 2 000 面天线，面对这种情况，传统方案只能重新开发一套针对扩建后计算规模的相关器；但是，图形处理器相关器只需根据新的计算任务，适当添加 F-engine 模块和 X-engine 模块的节点数量即可，并配备足够的交换机。

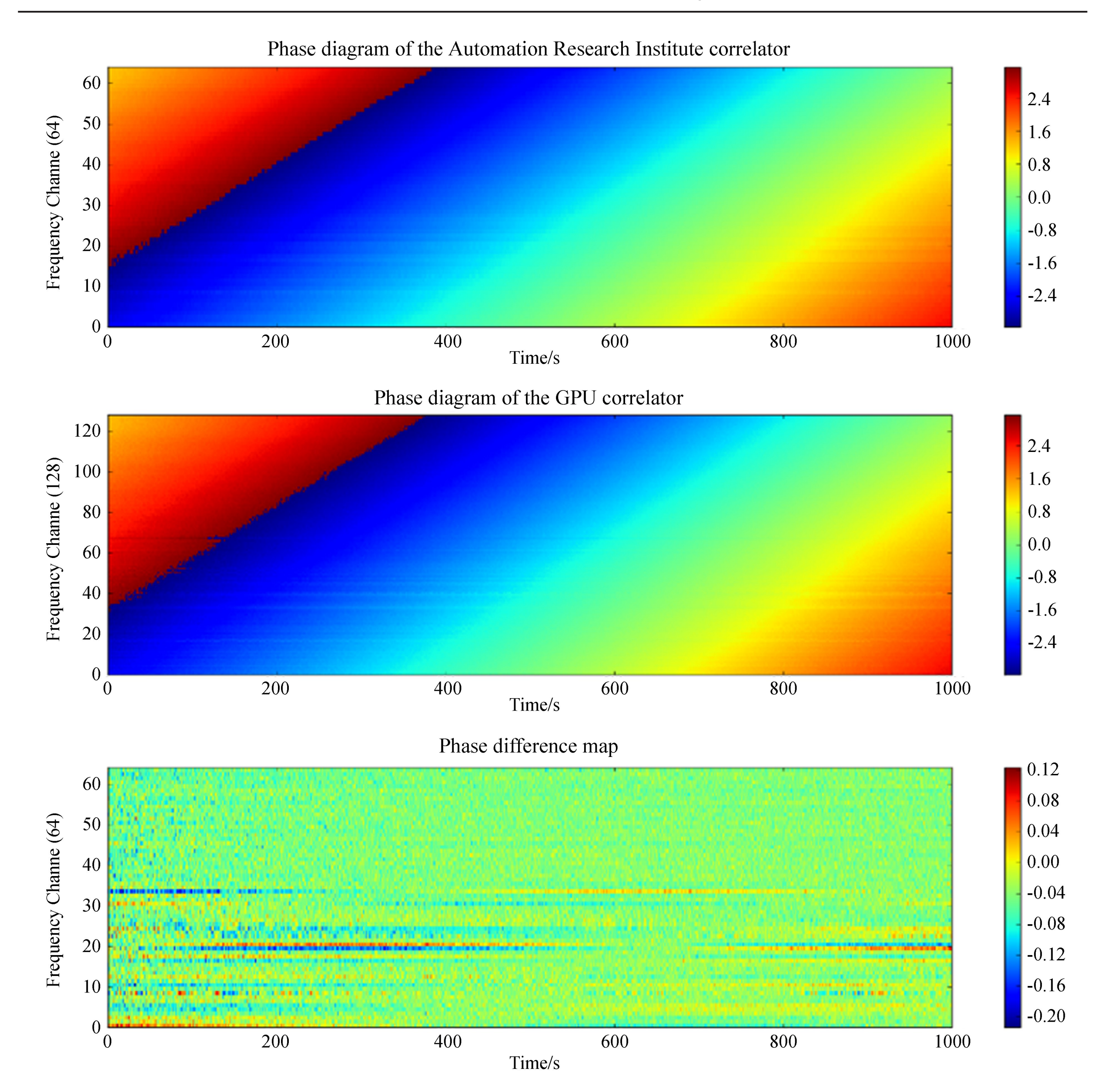


图 6 自动化所研制的相关器相位图(上)；图形处理器相关器相位图(中)；相位差图(下)

Fig. 6 The phase diagram of the correlator designed by the Automation Research Institute (top); The phase diagram of GPU correlator designed by our group (middle); The phase difference between the above two correlators (bottom)

除具备极好的可拓展性之外，图形处理器相关器还具有研发周期短的特性。相对于传统方案的相关器，图形处理器相关器所需的硬件直接采购，不需要重新开发；而对于软件部分，针对不同图形处理器以及不同参数的情况，只需对程序作适当优化或者对相应参数做修改即可。

此外，图形处理器相关器相对于传统方案的相关器来说，部署也很简单。只需要将几台服务器通过网络交换机组建一个集群；而传统方案的相关器则不然，由于一台机器的计算能力有限，所以一般情况下需多台机器，而每两台机器之间需要通过连线进行数据交换，过程极其繁杂。

4 讨 论

该图形处理器相关器较之传统方案的基于现场可编程门阵列的相关器，具有开发周期短、可扩展性强、部署简单等诸多优势。

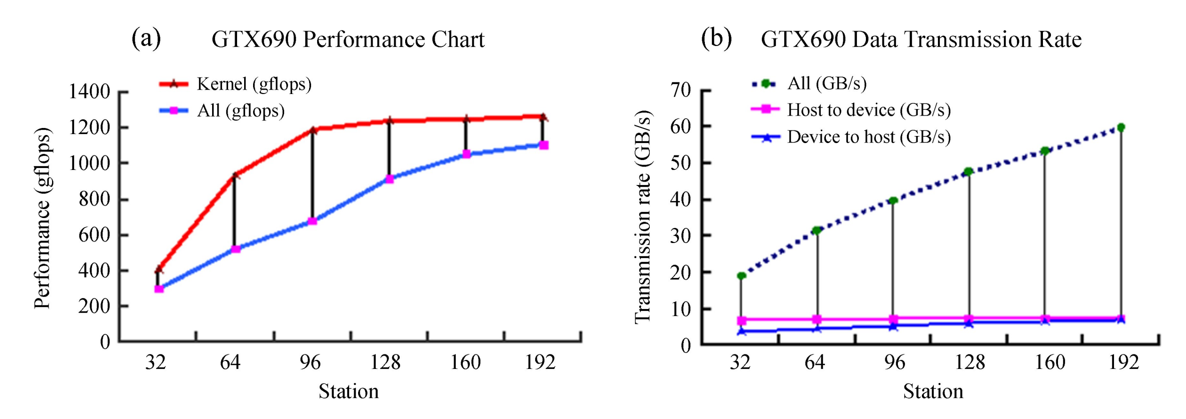


图 7 GTX690 在不同 STATION(天线个数)时的计算性能曲线图(a)和数据传输速率曲线图(b)

Fig. 7 The calculated performance chart of the GTX690 with different numbers of antennas (a) and the data transmission rate (b)

此外，在实验中还测试了该图形处理器相关器的线性度。相关器线性度用于衡量一个相关器自身性能的好坏，它表示相关器输入信号功率与计算输出结果(换算成功率)的一个线性范围。图 8 是该图形处理器集群相关器的线性度。

从图 8 可知，该相关器线性范围约在-12 dBm 到 6 dBm，即输入信号功率在该范围内，相关器计算结果可靠。

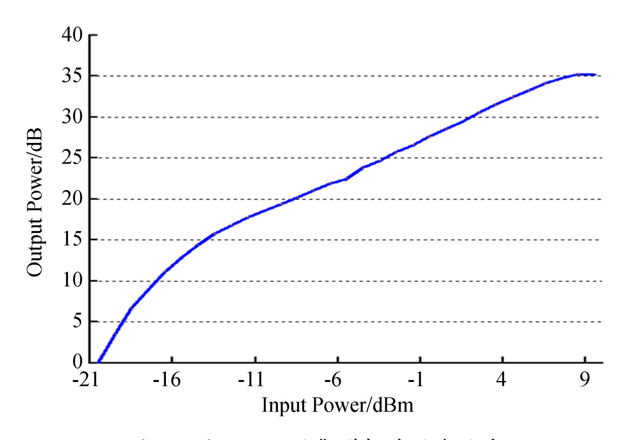


图 8 图形处理器集群相关器线性度

Fig. 8 The linearity of the GPU cluster correlation

5 总结与展望

本文基于图形处理器集群，针对大型射电干涉阵研发的一套扩展性极强的通用相关器并应用于“天籁计划”项目。在前期实验基础上，研制了可处理 32 路输入信号的图形处理器相关器系统。首先设计了一套基于图形处理器集群的通用相关器，该相关器采用分频分布式计算模式，结合硬件与软件编程，具备完美的可拓展性；然后，根据“天籁计划”一期的相关需求，包括数据传输压力与数据计算量等，详细讨论了图形处理器相关器各模块的功能以及实现；最后，对图形处理器相关器进行了性能测试并对其作了简单的讨论。在此前的实验中，只是简单地测试了单图形处理器的实际性能，而在本实验中，采用图形处理器集群，实现了“天籁计划”一期的数据的正确实时处理。对于项目一期规模，96 个双极化天线，该图形处理器集群相关器实际的计算性能为 35.6TFLOPS(约理论峰值性能的 46%)，该规模的计算任务是单图形处理器无法完成的。不过，目前该相关器系统的峰值性能利用率不高，主要原因是硬件采购针对规模为 128 个双极化天线的计算需求进行，图形处理器内核函数的性能利用率太低，所以图形处理器集群的计算能力未达饱和。

在后面的研究工作中，将对该图形处理器集群相关器模型作进一步优化，比如优化图形处理器集群中的数据传输，针对图形处理器集群优化内核函数等，使其实际的计算性能提升到 63TFLOPS，理论峰值利用率提高到 70%左右，以便为“天籁计划”的后续工作做准备。

参考文献：

[1] Harris C J. A parallel model for the heterogeneous computation of radio astronomy signal correlation [D]. Australia: the University of Western Australia, 2009.

[2] 陈学雷. 暗能量的射电探测天籁计划简介［J］. 中国科学：物理学 力学 天文学, 2011, 41(12)：1358-1366.
Chen Xuelei. Radio detection of dark energy-the Tianlai project［J］. Scientia Sinica：Physica, Mechanica & Astronomica, 2011, 41(12)：1358-1366.

[3] Clark M A, La Plante P C, Greenhill L J. Accelerating radio astronomy cross-correlation with graphics processing units［J］. International Journal of High Performance Computing Applications, 2013, 27(2)：178-192.

[4] 田海俊, 徐洋, 陈学雷, 等. 射电干涉阵 GPU 相关器的研究初探［J］. 天文研究与技术——国家天文台台刊, 2014, 11(3)：209-217.
Tian Haijun, Xu Yang, Chen Xuelei, et al. A preliminary study on GPU-based correlators for a radio interferometer array［J］. Astronomical Research & Technology——Publications of National Astronomical Observatories of China, 2014, 11(3)：209-217.

[5] 黄锦增, 陈虎, 赖路双. 异构 GPU 集群的任务调度方法研究及实现［J］. 计算机技术与发展, 2012, 22(5)：32-36.
Huang Jinzeng, Chen Hu, Lai Lushuang. Research and implementation of task schedule method on heterogeneous GPU cluster［J］. Computer Technology and Development, 2012, 22(5)：32-36.

[6] 张舒, 褚艳利, 赵开勇, 等. GPU 高性能运算之 CUDA［M］. 北京：中国水利水电出版社, 2009.

[7] 陈天超, 冯百明. 单精度浮点数累加和误差研究［J］. 计算机应用, 2013, 33(6)：1531-1533+1539.
Chen Tianchao, Feng Baiming.［J］. Research on error accumulative sum of single precision floating point［J］. Journal of Computer Applications, 2013, 33(6)：1531-1533+1539.

A Research on the ROACH2-GPU-Cluster-based Correlator
——The Design and Implementation of an X-engine Module

Wang Qunxiong[1,2], Niu Chenhui[2,3], Tian Haijun[1,2], Wu Fengquan[2], Li Jixia[2], Chen Xuelei[2], Hao Jie[4]
(1. China Three Gorges University, Yichang 443002, China, Email：1276303919@ qq.com; 2. National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China; 3. Central China Normal University, Wuhan 430079, China; 4. Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)

Abstract：As radio interference technology continues to improve, the scale of interferometric array becomes larger and larger. Its observation capacity also gradually increases. Yet real-time processing of big data becomes problematic. To tackle this kind of problem, this article takes the radio interferometer correlator's need of data computing and transmission, and the characteristics of the radio interferometric array signal into consideration and develops a set of generic correlator based on GPU cluster for the data processing work of "TianLai" project. First of all, considering radio signal's characteristics of correlation calculation, computing tasks are assigned to different GPU nodes according to their frequency bands, and the network load on each node is properly balanced; then these tasks are completed by the corresponding nodes and the results are sent to the storage nodes in real time; finally, the whole system is deployed with reference to the data processing scheme of the GPU cluster correlator, and a performance test is carried out based on the first stage requirements of "TianLai" project. According to the results, the node computing performance of the cluster correlator has been speeded up：it is around 46% of the theoretical peak performance. Compared with the traditional correlator, the GPU-cluster-based correlator is superior owing to its short development cycle, strong scalability, simple deployment and other advantages.

Key words：Radio interferometer; FX Correlator; GPU Cluster; Real-time data processing; Frequency dividing calculation